- **Task 1.2. Blocking Techniques - Put the results of all the experimental setups you have tried (different attributes, different data form of attribute, … etc.) and present which one you used for the final output.**

```
------ block on Title ------
0 Managing My Life: My Autobiography     409 Managing My Life: My Autobiography
8 Jenny Pitman: The Autobiography        1431 Jenny Pitman: The Autobiography
11 Call Me Anna: The Autobiography of Patty Duke         100 Call Me Anna: The Autobiography of Patty Duke
13 Chasing the Wind: The Autobiography of Steve Fossett          1152 Chasing the Wind: The Autobiography of Steve Fossett
22 Mad, Bad & Dangerous to Know: The Autobiography       192 Mad, Bad & Dangerous to Know: The Autobiography
There are 3484 pairs with blocking.
There are 40 matched pairs presented in dev set.
reduction_ratio: 0.00023729952055825054
pairs_completeness: 0.5970149253731343
pair_quality: 0.011481056257175661
------ block on Title[:3] ------
194 Man: An Autobiography        2625 Man Without A Face / Edition 1
194 Man: An Autobiography        1161 Man Of Everest - The Autobiography Of Tenzing
194 Man: An Autobiography        377 Man Who Lives in Paradise: Autobiography of A. C. Gilbert with Marshall McClintock
194 Man: An Autobiography        1887 Man and Woman, War and Peace 1941-1951: A Dual Autobiography
194 Man: An Autobiography        3568 Man Who Tried: An Autobiography
There are 759133 pairs with blocking.
There are 63 matched pairs presented in dev set.
reduction_ratio: 0.05170548132604661
pairs_completeness: 0.9402985074626866
pair_quality: 8.298941028778884e-05


------ block on ISBN13 ------
0 Managing My Life: My Autobiography 9780340728567       409 Managing My Life: My Autobiography 9780340728567
10 A Man Called White: The Autobiography of Walter White 9780820316987   769 A Man Called White: The Autobiography of Walter
White / Edition 1 9780820316987
291 Bertolt Brecht: A Literary Life (Biography and Autobiography)        1315 Recovering Literature's Lost Ground: Essays in
American Autobiography
649 Against The Evil Tide - An Autobiography     1315 Recovering Literature's Lost Ground: Essays in American Autobiography
2676 Autobiography of Andrew T. Still            1315 Recovering Literature's Lost Ground: Essays in American Autobiography
There are 1239 pairs with blocking.
There are 37 matched pairs presented in dev set.
reduction_ratio: 8.438981227659942e-05
pairs_completeness: 0.5522388059701493
pair_quality: 0.02986279257465698
------ block on Title[:3] + 1st Author[:3] ------
2 Betty Boothroyd: Autobiography         2851 Betty: The Autobiography
2 Betty Boothroyd: Autobiography         2541 Betty Boothroyd Autobiography
559 Betty: The Autobiography     2851 Betty: The Autobiography
559 Betty: The Autobiography     2541 Betty Boothroyd Autobiography
563 Betty: The Autobiography     2851 Betty: The Autobiography
There are 9543 pairs with blocking.
There are 54 matched pairs presented in dev set.
reduction_ratio: 0.0006499854548471254
pairs_completeness: 0.8059701492537313
pair_quality: 0.005658597925180761
```

Trail1：Title
Trail2：Title[:3] (the first three characters of the Title)
Trail3：ISBN
Trail4：Title[:3] + 1st Author[:3] (the first three characters of the Title plus the first three characters of the first author's name)

The 4th trail's performance is the best. And the logic seems right. Therefore, I chose this method to perform blocking. (reduction ratio < 0.1, pair completeness > 0.7)

- **Task 1.3. Entity Linking - Put all the experimental results you have tried (jaro winkler similarity, … etc.) and present which one you used for the final output.**

```python
def name_string_similarity_1(r1, r2):
    s1 = r1.Title
    s2 = r2.Title
    return rltk.jaro_winkler_similarity(s1, s2)

def name_string_similarity_2(r1, r2):
    s1 = r1.ISBN
    s2 = r2.ISBN
    if s1 == s2:
        return 1
    return 0

def name_string_similarity_3(r1, r2):
    for n1, n2 in zip(sorted(r1.Title), sorted(r2.Title)):
        if rltk.levenshtein_distance(n1, n2) > min(len(n1), len(n2)) / 3:
            return 0
    return 1
```

```python
def rule_based_method(r1, r2):
    score_1 = name_string_similarity_1(r1, r2)
    score_2 = name_string_similarity_2(r1, r2)
    score_3 = name_string_similarity_3(r1, r2)

    total = 0.8 * score_1 + 0.1 * score_2 + 0.1 * score_3

    # return two values: boolean if they match or not, float to determine confidence
    return total > MY_TRESH, total
```

I tried jaro_winkler_similarity, levenshtein_distance, and the normal comparison. The final method = 80% jaro_winkler_similarity + 10% levenshtein_distance + 10% True/False normal comparison.

```
trial.f_measure
```

```
Last executed at 2022-09-19 20:55:34 in 14ms
```

```
0.823529411764706
```

- **Task 2.1. KG Construction - Explain your KG design.**

I chose ID as URI. (unique) The number of valid predictions pairs is 923 in my case. I added all the values into my graph and defined properties and data type using Schema. The final model is too large to be visualized, so I chose a single part of it to be visualized using the provided tool.